Disk-to-tape performance tuning

CASTOR workshop
28-30 November 2012

Eric Cano
on behalf of CERN IT-DSS group

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Network contention issue

- Not all disk servers at CERN have 10Gb/s interfaces (yet)
- Output on NIC in disk servers is a contention point
- Tape servers equally compete with other streams
- Tape write speed now fine with buffered tape marks, yet…
- A tape server's share can drop below 1MB/s
  - 100s of simultaneous connection on the same disk server
- With data taking, this can lead to tape server starvation, spreading on all castor instances

First solution: software level

- We turned on scheduling, which allowed capping the number of clients per disk server to a few 10s
- Cannot go lower as a client can be slow as well, and we want to keep the disk server busy (from bandwidth starvation to transfer slot starvation)
- We need a bandwidth budgeting system tolerating a high number of sessions, yet reserving bandwidth to tape servers

Second solution: system level

- Using Linux kernel traffic control
- Classify outbound traffic in disk servers between favoured (tape servers) and background (the rest)
- Still in test environement
- The tools:
  - tc (qdisc, class, filter)
  - ethtool (-k, -K)
- Some technicalities:
  - with tcp segmentation offload kernel sees too big packets, fails to shape traffic

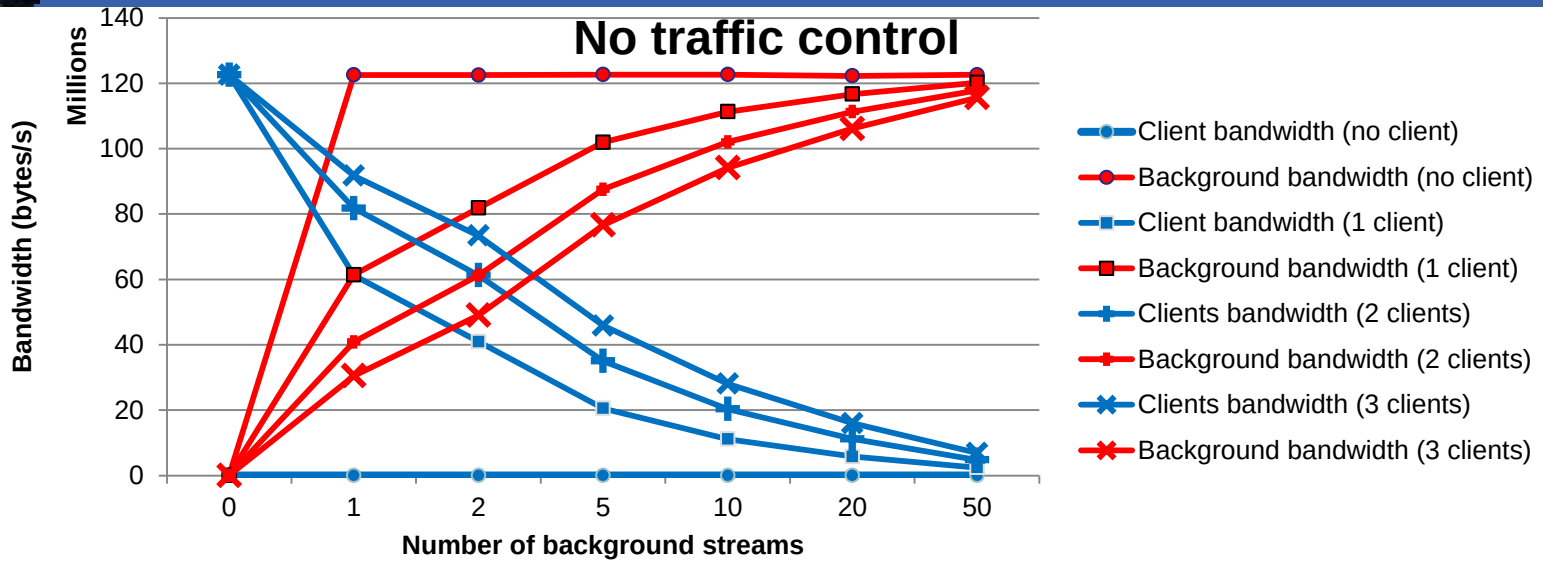
Shaping details

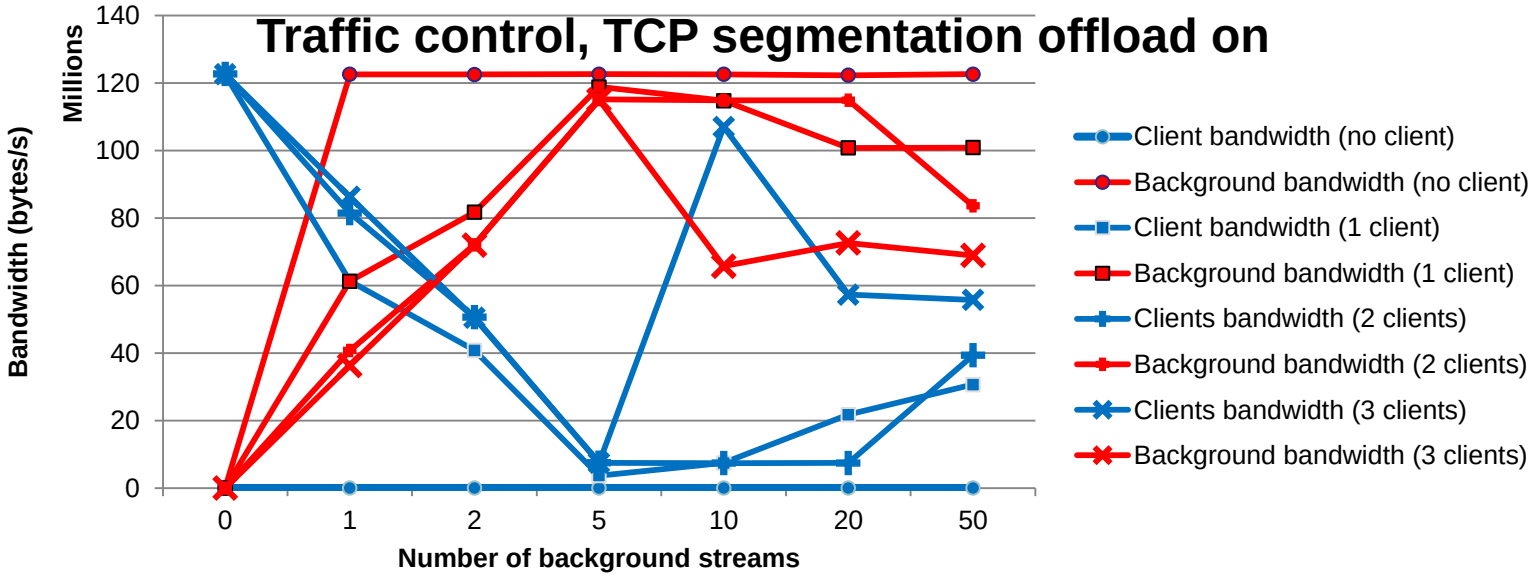
- 3 priority queues by default:
  - Interactive, best effort, bulk
- Retain the mechanism (using tc qdisc prio)
- Within the best effort queue, classify and prioritize outbound traffic (filter)
  - Tape servers, but also
  - ACK packets, helping incoming traffic (all big streams are one-way)
  - ssh, preventing non-interactive ssh (wassh) from timing out
- Token bucket filter (tbf) and hierarchical token bucket (htb) did not give expected result
- Using class based queuing (cbq)
- Keep 90/10 mixing between low and high priority classes to keep all connections alive

The gory details

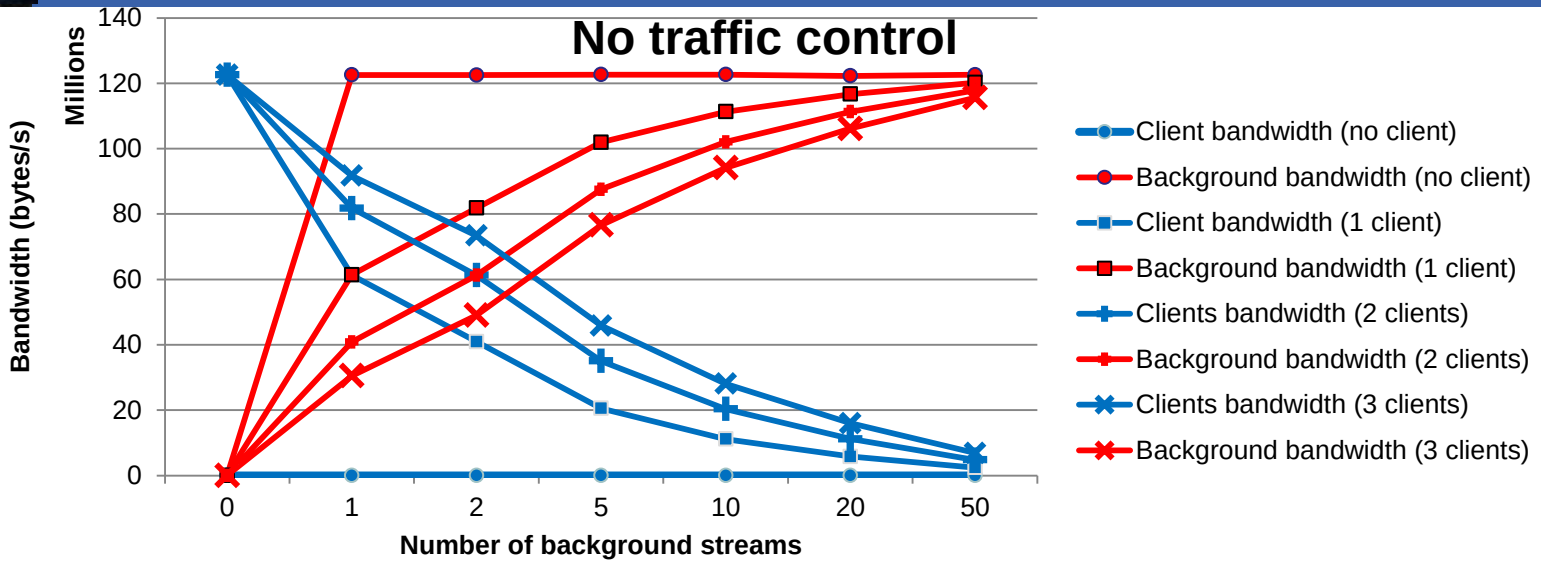

```
#!/bin/bash
# Turn off TCP segmentation offload: Kernel sees the details of the packets routing
/sbin/ethtool -K eth0 tso off
# Flush the existing rules (gives an error when there are none)
tc qdisc del dev eth0 root 2> /dev/null
# Duplication of the default kernel behavior
tc qdisc add dev eth0 parent root handle 10: prio bands 3 priomap 1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
# Creation of the class based queuing
tc qdisc add dev eth0 parent 10:1  handle 101: cbq bandwidth 1gbit avpkt 1500
tc class add dev eth0 parent 101: classid 101:10 cbq weight 90 split 101: defmap 0 bandwidth 1gbit \
       prio 1 rate 900mbit maxburst 20 minburst 10 avpkt 1500
tc class add dev eth0 parent 101: classid 101:20 cbq weight 10 split 101: defmap ff bandwidth 1gbit\
       prio 1 rate 100mbit maxburst 20 minburst 10 avpkt 1500
# Prioritize ACK packets
tc filter add dev eth0 parent 101: protocol ip prio 10 u32 match ip protocol 6 0xff\
       match u8 0x05 0x0f at 0 match u16 0x0000 0xffc0 at 2 match u8 0x10 0xff at 33\
       flowid 101:10
# Prioritize SSH packets
tc filter add dev eth0 parent 101: protocol ip prio 10 u32 match ip sport 22 0xffff flowid 101:10
# Prioritize network ranges of tape servers
tc filter add dev eth0 parent 101: protocol ip prio 10 u32 match ip dst <Network1>/<bits1> flowid 101:10
tc filter add dev eth0 parent 101: protocol ip prio 10 u32 match ip dst <Network2>/<bits2> flowid 101:10
<etc..>
```

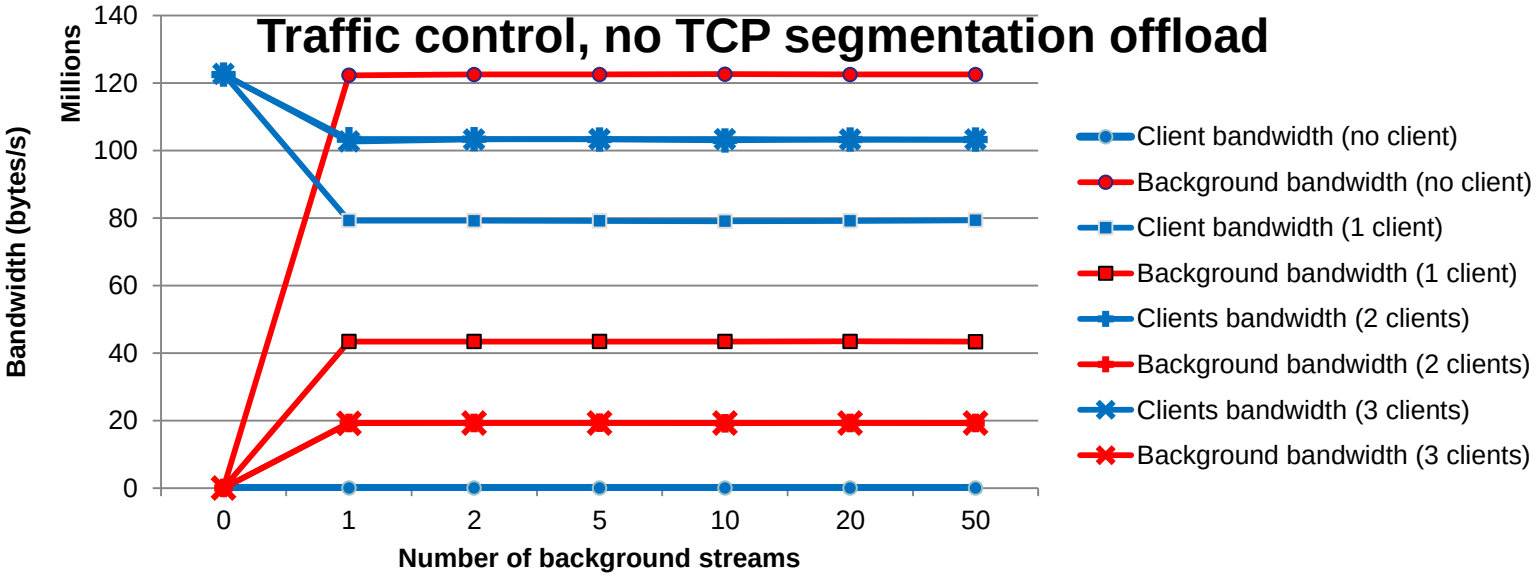
Traffic control results





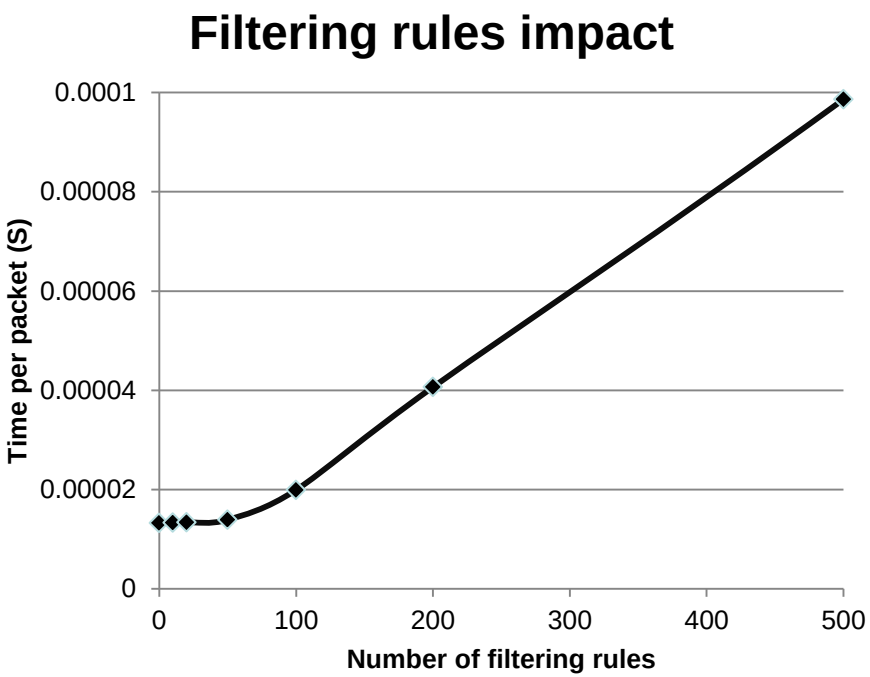
Traffic control results





Cost of filtering rules

- Time per packet linear with number or rules for n>100
- Average rule processing time: ~200-250 ns
- Packet time: (1500b/1Gb/s ~12 µs
- => 48-60 rules maximum



- Test system (for reference):
  - Intel Xeon E51500 @ 2.00GHz
  - Intel 80003ES2LAN Gigabit (Copper, dual port, 1 used)
  - Linux 2.6.18-274.17.1.el5
- ~122 tape servers in production: per-host rules won't fit
- Per network service appropriate at CERN (11 IP ranges)

Conclusions

- Traffic shaping has been well understood in test environment, and prioritizes work appropriately
- Tape traffic will remain on top in any disk traffic conditions
- Other traffic will not be brought to 0 and timeout
- Bidirectional traffic should be helped too
- Yet, filtering rules budget is small
  - Ad-hoc rules necessary (will work for CERN)
  - No easy one-size-fits-all tool (showqueue/cron based for exemple)